Search Engine

Blue Online - Lecture 14: Trie

Tóm tắt đề bài

Tóm tắt đề bài

Lời giới thiệu: Các công cụ tìm kiếm hiện nay đều có chức năng auto-complete. Với chức năng này, mình không cần phải gõ toàn bộ từ khóa, khi chỉ mới gõ 1 phần của từ khóa là công cụ sẽ tự động giới thiệu các kết quả phù hợp.

Đề bài cho mình 1 tập dữ liệu gồm nhiều xâu, mỗi xâu đều sẽ có 1 con số nguyên đại diện cho độ ưu tiên. (Con số càng lớn, độ ưu tiên càng lớn).

Nhiệm vụ của mình sẽ là với mỗi từ khóa, in ra độ ưu tiên lớn nhất của các xâu mà nhận từ khóa tìm kiếm là tiền tố.

Mô tả input/output

Input

- Dòng đầu tiên gồm n và q, số lượng xâu trong tập dữ liệu và số lượng từ khoá cần tìm kiếm.
- n dòng tiếp theo: mỗi dòng lần lượt là một xâu trong tập dữ liệu và một số nguyên đại diện cho độ ưu tiên của xâu đó (w).
- q dòng tiếp theo: mỗi dòng là một từ khoá.

Giới hạn:

- $1 \le n, w, len(s), len(t) \le 10^6$
- $1 \le q \le 10^5$
- Tổng độ dài các xâu trong tập dữ liệu $\le 10^6$
- Tổng độ dài các từ khóa $\le 10^6$

Output

- Với mỗi từ khóa, xuất ra độ ưu tiên lớn nhất của xâu có tiền tố là từ khóa đó
- Nếu không có từ nào có tiền tố là từ khóa đó, thì in -1.

Giải thích ví dụ

Ví dụ

Input

```
2 1
hackerearth 10
hackerrank 9
hacker
```

Output

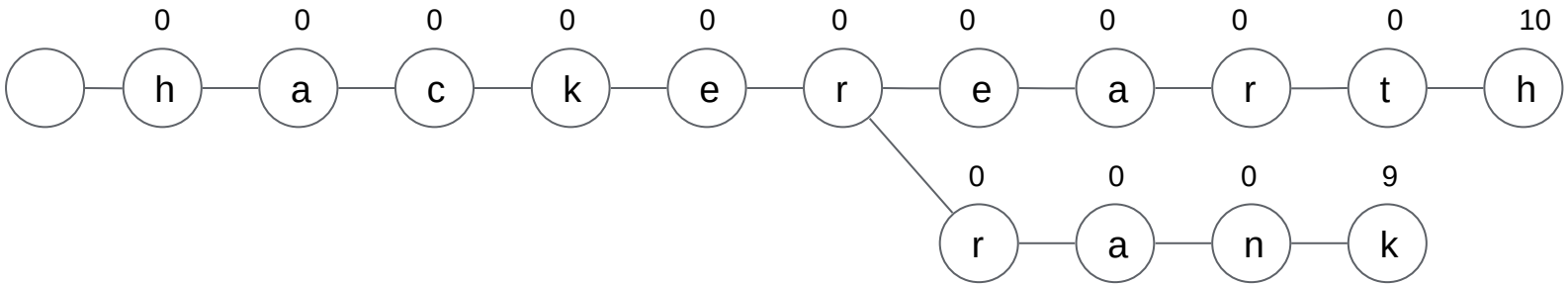
```
10
```

- Tập dữ liệu có 2 xâu: **hackerearth** và **hackerrank**.
- Với từ khoá **hacker**, cả hai xâu trong tập dữ liệu đều thoả, nhưng xâu **hackerearth** có độ ưu tiên lớn nhất là **10**.
- Đáp án được in ra sẽ là **10**.

Hướng dẫn giải

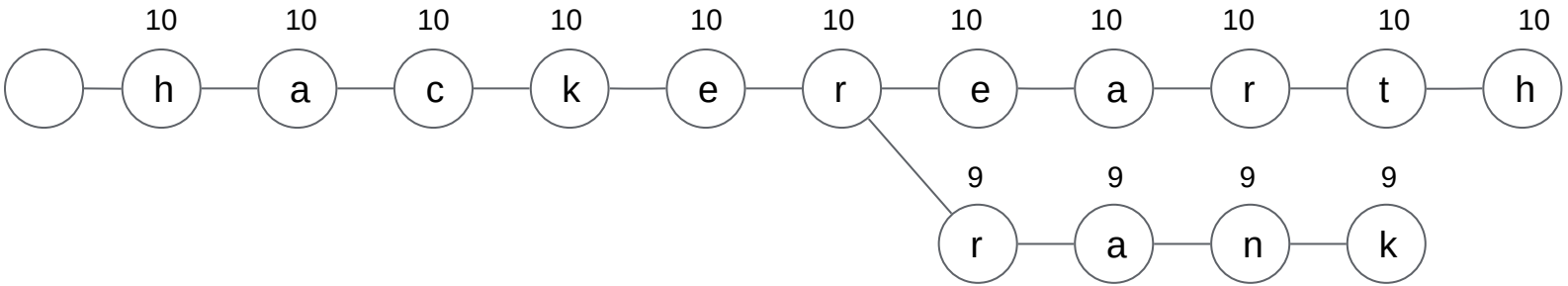
Ý tưởng cơ bản

- Thay vì lưu countWord = 1 để đánh dấu node đó là kết thúc của 1 xâu, mình lưu countWord = độ ưu tiên của xâu đó.
- Khi đó, với mỗi từ khóa x, để tìm từ có độ ưu tiên cao nhất, mình phải duyệt qua trên cây đến tất cả xâu có tiền tố x.
- Ví dụ: Tìm độ ưu tiên lớn nhất ứng với từ khóa “hacker”: duyệt đến node cuối của xâu “hackerearth” và xâu “hackerrank” mới lấy được độ ưu tiên.
- Độ phức tạp: **O(n*string_length + q*string_length*n)**.
- **Bị quá thời gian.**



Cải tiến

- Với mỗi từ khóa mình chỉ quan tâm đến xâu trong tập dữ liệu có từ khóa này làm tiền tố và có độ ưu tiên lớn nhất.
- Thay vì chỉ lưu độ ưu tiên ở node cuối của xâu, mình sẽ lưu độ ưu tiên lớn nhất ở tất cả node của xâu. Mỗi node sẽ có thêm thuộc tính **maxValue**.
- Khi tìm độ ưu tiên lớn nhất của 1 từ khóa nào đó, chỉ cần duyệt tới node cuối cùng của từ khóa đó.
- Ví dụ: Để tìm độ ưu tiên lớn nhất ứng với từ khóa “hacker”, chỉ cần duyệt từ node gốc tới node cuối của từ khóa đó (node “r”).



Mã giả

Mã giả

```
class Node:
  child = [NULL] * 26
  maxValue = -1

def addWord(root, word, value):
  temp = root
  for ch in word:
    pos = ch - 'a'
    if temp.child[pos] == NULL:
      temp.child[pos] = Node()
    temp = temp.child[pos]
    temp.maxValue =
max(temp.maxValue, value)
```

```
def getMaxValue(root, word):
  temp = root
  for ch in word:
    pos = ch - 'a'
    if temp.child[pos] == NULL:
      return -1
    temp = temp.child[pos]
  return temp.maxValue
```

Mã giả

```
Read n, q
root = Node()
for i = 0 to N-1:
  Read s, w
  addWord(root, s, w)

for i = 0 to q-1:
  Read t
  print(getMaxValue(root, t))
```

Độ phức tạp: **O(n*string_length + q*string_length)**

Thank you